



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/771,837

02/03/2004

Weiwen Liu

MS1-1834US

9735

22801 7590 07/16/2007  
LEE & HAYES PLLC  
421 W RIVERSIDE AVENUE SUITE 500  
SPOKANE, WA 99201

EXAMINER

ANYA, CHARLES E

ART UNIT

PAPER NUMBER

2194

MAIL DATE

DELIVERY MODE

07/16/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

## Office Action Summary

Application No.

10/771,837

Applicant(s)

LIU ET AL.

Examiner

Charles E. Anya

Art Unit

2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 30 April 2007.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1,3-11,13-28,30-37 and 39-54 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,3-11,13-28,30-37 and 39-54 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.

- WILLIAM THOMSON  
SUPERVISORY PATENT EXAMINER
- 4) ☒ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_.

### DETAILED ACTION

1. Claims 1,3-11,13-28, 30-37, and 39-54 are pending in this application.

### ***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. **Claims 1,3-6,8,10,15,16,23-28,30,32,34,36,41 and 48-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al.**

As to claim 1, Liang teaches a computer readable medium comprising computer-program instructions executable by a processor and implementing instructions for: a runtime hosting interface comprising a host abstraction interface (HAI), the HAI corresponding to execution environment abstraction(s) supported by a host application (VMPI 215/Profiler 220 page 2 paragraphs 0024-026, page 8 paragraphs 0183/0184, "...JVMPI..." page 11 paragraph 0258), at least one specific interface or object corresponding to a specific one HAI being accessible by a runtime during execution of runtime managed code and responsive to an action or event associated with an identified one HAI, the HAI providing an interface for the runtime to configure host

Art Unit: 2194

execution environment parameters and/or notify the host application of a runtime event (“...notifies...” page 8 paragraphs 0183/0184, “...NotifyEvent...” page 9 paragraphs 0207-0227, page 11 paragraphs 0268); the host application or the runtime negotiating, which will perform certain functions (“...two-way function call interface...” page 2 paragraph 0026); the runtime allocating memory via at least one specific interface or object implemented by the host application (page 2 paragraph 0012, VMPI 215 page 3 paragraph 0046); and the runtime notifying when a task has been moved to and/or from a runnable state (“...notify...” page 8 paragraph 0183, “...Event Notification...” page 9 paragraphs 0208-0216).

Liang is silent with reference to the runtime obtaining additional information during process execution (“...calling back...” page 2 paragraph 0026).

Benayon teaches the runtime obtaining additional information during process execution (“...calling back...” Col. 3 Ln. 23 – 31).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Liang with the teaching of Benayon because the teaching Benayon would improve the system of Liang by providing a method of determining the availability of minimum size of heap memory and allocating the heap extension to a user application (Col. 3 Ln. 23 – 31).

4. As to claim 3, Liang teaches the computer-readable medium as recited in claim 1, wherein the one or more execution environment abstractions correspond to management services for memory, threads/tasks, I/O completion, synchronization,

runtime entry/exit notification, security context, impersonation, runtime configuration, customized assembly loading, host protection, garbage collection (GC), debugging, and/or executable code service abstractions ("...JVMPI..." pages 7/8 paragraphs 0095-0180).

5. As to claim 4, Liang teaches the computer-readable medium as recited in claim 1, wherein the runtime hosting interface further comprises a runtime interface (RI) for use by the host application to configure operations of the runtime, notify the runtime of an event, and/or to obtain additional information during host application process execution (page 11 paragraphs 0247-0251/0265-0267, page 14 paragraphs 0324/0325).

6. As to claim 5, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to identify host application implemented ones of the HAI or associated object(s) for subsequent calling responsive to an action or event associated with an identified one of the respective execution environment abstractions (page 9 paragraphs 0201-0206).

7. As to claim 6, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to request the host application to perform a memory allocation (JVMPI\_EVENT\_OBJECT\_ALLOC page 19 paragraphs 0433/0434).

8. As to claim 8, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to create a new thread/task via the HAI (page 11 paragraphs 0247/0265-0267).

9. As to claim 10, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application that a task cannot be moved to a different physical OS thread and cannot have execution of the task blocked for a specified window of time (page 19 paragraphs 0445-0047).

10. As to claim 15, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to modify an object identified by an interface of the HAI ("...priority..." page 12 paragraph 0268).

11. As to claim 16, Liang teaches the computing device as recited in claim 15, wherein the object is a task priority ("...priority..." page 12 paragraph 0268).

12. As to claim 23, Benayon teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to supply the host application with a runtime implemented callback, the runtime implemented callback for

Art Unit: 2194

invoking by the host application when an asynchronous I/O operation completes

("...second function..." Col. 4 Ln. 49 – 53, Col. 6 Ln. 18 – 19).

13. As to claim 24, Benayon teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to supply a runtime implemented callback to the host application, the runtime implemented callback to be invoked by the host application when an asynchronous I/O operation completes, the runtime implemented callback being used by the runtime to provide custom state information to the host application ("...callback function..." Col. 3 Ln. 25 – 31, "...second function..." Col. 4 Ln. 49 – 53).

14. As to claim 25, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to: generate a task (CreateSystemThread page 11 paragraphs 0266/027); and specify one or more synchronization objects for the task to substantially ensure host application knowledge of a lock on the task, the one or more synchronization objects comprising a critical section, a manual and/or auto-reset event, a semaphore, a reader/writer lock, and/or a task monitor ("JVMPI\_EVENT\_MONITOR..." page 18 paragraphs 0413-0421).

15. As to claim 26, Liang teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application of one or more runtime interfaces allowing the host application to notify the runtime of

events and/or to obtain additional information during host application process execution (“...RequestEvent...” page 14 paragraphs 0324/0325).

16. As to claim 27, see the rejection of claim 1 above.

17. As to claim 28, Liang teaches a computing device for enhanced runtime hosting, the computing device comprising: means for identifying, by a runtime one or more execution environment abstractions implemented by a host application, the host application for hosting the runtime (page 7 paragraphs 0095 – 0144, page 8 paragraphs 0145 – 0183); during execution of runtime managed code and responsive to an action or event associated with an identified one of the respective execution environment abstractions, means for the runtime to interface with specific ones of the execution environment abstractions (page 7 paragraphs 0095 – 0144, page 8 paragraphs 0145 – 0183); the HAI providing an interface for the runtime to configure host execution environment parameters and/or notify the host application of a runtime event (“...Event Notification...” page 9 paragraphs 0207 – 0226), wherein the execution environment abstractions correspond to memory management, threads/tasks, I/O completion, synchronization, runtime entry/exit notification, security context, impersonation, runtime configuration, executable service code abstractions, and/or garbage collection (GC) (“...dynamically managing storage allocation...” page 2 paragraph 0012, “Allocation and Garbage Collection Events...” page 3 paragraphs 0038 – 0040) and means for the host application or the runtime negotiating, which will perform certain functions (“...two-way



Art Unit: 2194

function call interface...” page 2 paragraph 0026); means for the runtime allocating memory via at least one specific interface or object implemented by the host application (page 2 paragraph 0012, VMPI 215 page 3 paragraph 0046); means for the runtime notifying when a task has been moved to and/or from a runnable state (“...notify...” page 8 paragraph 0183, “... Event Notification...” page 9 paragraphs 0208-0216).

Liang is silent with reference to an interface of the HAI provides the runtime with a pointer to an object associated with the interface, the object for calling by the runtime responsive to a specified event or criteria and means for the runtime obtaining additional information during process execution.

Benayon teaches an interface of the HAI provides the runtime with a pointer to an object associated with the interface, the object for calling by the runtime responsive to a specified event or criteria (Col. 5 Ln. 60 – 67) and means for the runtime obtaining additional information during process execution (“...calling back...” page 2 paragraph 0026).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Liang with the teaching of Benayon because the teaching Benayon would improve the system of Liang by providing a method of determining the availability of minimum size of heap memory and allocating the heap extension to a user application (Benayon Col. 3 Ln. 23 – 31).

18. As to claim 30, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions comprise means for interfacing with an

object associated with the host application, the runtime interfacing with the object responsive to a specified event or criteria that occurs during host application execution (page 11 paragraphs 0247-0251/0265-0267, page 14 paragraphs 0324/0325).

19. As to claim 31, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions comprise means for the host application to configure operations of the runtime, notify the runtime of an event, and/or to obtain additional information during host application process execution (“...RequestEvent...” page 14 paragraphs 0324/0325).

20. As to claim 32, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to request a memory allocation (JVMPI\_EVENT\_OBJECT\_ALLOC page 19 paragraphs 0433/0434).

21. As to claim 34, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions comprise means for the runtime to create a new thread/task (page 11 paragraphs 0247/0265-0267).

22. As to claim 36, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to notify the host application that a task cannot be moved to a different physical OS

thread and cannot have execution of the task blocked for a specified window of time (page 19 paragraphs 0445-0447).

23. As to claim 41, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to adjust priority of a task associated with the host application (“...priority...” paragraph 0268).

24. As to claim 48, Benayon teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to indicate a runtime implemented callback to the host application, the runtime implemented callback for calling by the host application when an asynchronous I/O operation completes (“...second function...” Col. 4 Ln. 49 – 53, Col. 6 Ln. 18 – 19).

25. As to claim 49, Benayon teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to supply a runtime implemented callback to the host application, the runtime implemented callback for invoking by the host application when an asynchronous I/O operation completes, the runtime implemented callback giving the host application an opportunity to communicate custom state information to the runtime implemented callback (“...callback function...” Col. 3 Ln. 25 – 31, “...second function...” Col. 4 Ln. 49 – 53).

26. As to claim 50, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to: generate a task; and create one or more synchronization objects for the task to substantially ensure host application knowledge of a lock on the task, the one or more synchronization objects comprising a critical section, a manual and/or auto-reset event, a semaphore, a reader/writer lock, and/or a task monitor (“JVMPi\_EVENT\_MONITOR...” page 18 paragraphs 0413-0421).

27. As to claim 51, Liang teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to notify the host application of one or more runtime interfaces exposed by the runtime, the runtime interfaces for the host application to notify the runtime of an event and/or to obtain additional information during process execution (“...RequestEvent...” page 14 paragraphs 0324/0325).

28. As to claim 52, Benayon teaches a computer-readable medium as recited in claim 1, wherein an interface of the HAI provides the runtime with a pointer to an object associated with the interface, the object for calling by the runtime responsive to a specified event or criteria (Col. 5 Ln. 60 – 67).

**29. Claims 7 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pub. No. 20030056076 A1 to Cook et al.**

30. As to claim 7, Liang is silent with reference to the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to communicate a low memory notification from an OS to the host application, and/or inform the host application of consequences of failing a particular memory allocation via an HAI.

Cook teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to communicate a low memory notification from an OS to the host application, and/or inform the host application of consequences of failing a particular memory allocation via an HAI (Col. 5 Ln. 32 – 67, Col. 6 Ln. 1 – 7).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Cook because the teaching Cook would improve the system of Benayon and Liang by providing method of managing memory allocation by determining the availability of memory (Cook Col. 1 Ln. 39 – 47).

31. As to claim 33, see the rejection of claim 7 above.

**32. Claims 9,13,35 and 39 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pat. No. 7,007,269 B2 to Sluiman et al.**

33. As to claim 9, Liang is silent with reference to the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to reuse or pool a runtime-implemented portion of a task via the HAI.

Sluiman teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to reuse or pool a runtime-implemented portion of a task via the HAI (Col. 6 Ln. 15 – 20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Sluiman because the teaching Sluiman would improve the system of Benayon and Liang by providing a process of identifying a thread and allowing other trace elements to reference the thread through the transientThreadId (Sluiman Col. 12 Ln. 44-45).

34. As to claim 13, Sluiman teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to: provide a callback to the host application for notifying the runtime that a task has been moved to a different locale or a locale has changed for the task; and/or notify the host application, that a task

Art Unit: 2194

has been moved to a different locale or a locale has changed for the task (Col. 4 Ln. 39 – 44).

35. As to claims 35 and 39, see the rejection of claims 9 and 13 respectively.

**36. Claims 11 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pat. No. 6,915,457 B1 to Miller.**

37. As to claim 11, Liang is silent with reference to the computer-readable medium as recited in claim 1, wherein the HAI comprises: an interface for the runtime to indicate a callback to the host application, the callback for notifying the runtime when a task has been moved to a runnable or non-runnable state; and if the task has been moved to a non-runnable state, an interface to specify that the task is to be rescheduled as soon as possible by the host application.

Miller teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises: an interface for the runtime to indicate a callback to the host application, the callback for notifying the runtime when a task has been moved to a runnable or non-runnable state; and if the task has been moved to a non-runnable state, an interface to specify that the task is to be rescheduled as soon as possible by the host application (figure 10 Col. 15 Ln. 33 – 67, Col. 16 Ln. 1 – 13).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Miller because the teaching Miller would improve the system of Benayon and Liang by providing a process of restoring paths when it is determined that a application program has failed (Miller Col. 15 Ln. 33 – 34).

38. As to claim 37, see the rejection of claim 11 above.

**39. Claims 14 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pub. No. 20030167421 A1 to Klemm (Hereinafter referred Klemm'421).**

40. As to claim 14, Liang is silent the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to delay the host application abort of a task.

Klemm'421 teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to delay the host application abort of a task (Step 513 page 8 paragraph 0106).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Klemm'421 because the teaching Klemm'421 would improve the system of Benayon



Art Unit: 2194

and Liang by providing a process for detecting thread starvation (Klemm'421 page 7 paragraph 0102).

41. As to claim 40, see the rejection of claim 14 above.

**42. Claims 17-20 and 42-45 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pat. No. 5,909,580 to Crelier et al.**

43. As to claim 17, Liang is silent with reference to the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application that a task/thread is to leave the runtime into unmanaged code.

Crelier the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application that a task/thread is to leave the runtime into unmanaged code (Col. 16 Ln. 50 – 61).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Crelier because the teaching Crelier would improve the system of Benayon and Liang by providing a method of avoiding re-entry into a runtime library (Crelier Col. 16 Ln. 50 – 61).

Art Unit: 2194

44. As to claim 18, Crelier teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application that a task/thread is to reverse-leave the runtime into unmanaged code (Col. 16 Ln. 50 – 61).

45. As to claim 19, Crelier teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application that a task/thread is to enter the runtime from unmanaged code (Col. 16 Ln. 50 – 61).

46. As to claim 20, Crelier teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to notify the host application that a task/thread is to reverse-enter the runtime from unmanaged code (Col. 16 Ln. 50 – 61).

47. As to claim 42-45, see the rejection of claims 17-20 respectively.

**48. Claims 21,46,53 and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pat. No. 6,457,142 B1 to Klemm et al.**

Art Unit: 2194

49. As to claim 21, Benayon and Liang are silent with reference to the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to specify a maximum number of threads/tasks that will be available to service requests on one or more I/O completion ports.

Klemm teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to specify a maximum number of threads/tasks that will be available to service requests on one or more I/O completion ports (“...user-specified threshold...” Col. 6 Ln. 19 – 20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Klemm because the teaching Klemm would improve the system of Benayon and Liang by providing a generic software system that automatically detects and resolves many reliability and performance problems according to user specifications (Klemm Col. 2 Ln. 49 – 54).

50. As to claim 46, see the rejection of claim 21 above.

51. As to claim 53, Klemm teaches the computing device as recited in claim 28, wherein the execution environment abstractions further comprise means for the runtime to: queue a thread/task to a host application implemented thread pool; set a size of the host application implemented thread pool; and/or query the host application implemented thread pool (“...queue up...” Col.15 Ln. 20 – 21).

52. As to claim 54, see the rejection of claim 53 above.

**53. Claims 22 and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pub. No. 20040158589 A1 to Liang et al. in view of U.S. Pat. No. 6,816,956 B1 to Benayon et al. as applied to claims 1 or 28 above, and further in view of U.S. Pub. No. 20030093433 A1 to Seaman et al.**

54. As to claim 22, Benayon and Liang are silent with reference to the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to binding a handle to an I/O completion port of the host application.

Seaman teaches the computer-readable medium as recited in claim 1, wherein the HAI comprises an interface for the runtime to binding a handle to an I/O completion port of the host application (page 10 paragraphs 0168/0173).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the system of Benayon and Liang with the teaching of Seaman because the teaching Seaman would improve the system of Benayon and Liang by providing interfaces (General Movement Classes) for handling of data movement to and from a process (Seaman page 10 paragraph 0168).

55. As to claim 47, see the rejection of claim 22 above.

### ***Response to Arguments***

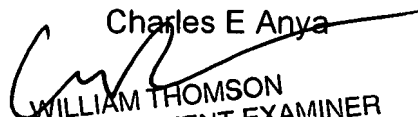
Applicant's arguments with respect to claims 1,3-11,13-28, 30-37, and 39-54 have been considered but are moot in view of the ground(s) of rejection.

### **Conclusion**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles E. Anya whose telephone number is (571) 272-3757. The examiner can normally be reached on M-F (8:30-5:00).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Charles E Anya  
  
WILLIAM THOMSON  
SUPERVISORY PATENT EXAMINER